УДК 575.8+51.76
DOI: 10.33184/bulletin-bsu-2020.2.10

АЛГОРИТМЫ ПОИСКА В ЗАДАЧАХ АНАЛИЗА НУКЛЕОТИДНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ С ЦЕЛЬЮ ОДНОЗНАЧНОЙ ИДЕНТИФИКАЦИИ ГЕНОМОВ

© О. Ю. Кирьянова[1]*, Л. У. Ахметзянова[1,2], И. М. Губайдуллин[1,2]

[1]Уфимский государственный нефтяной технический университет
Россия, Республика Башкортостан, 450064 г. Уфа, ул. Космонавтов, 1.

[2]Институт нефтехимии и катализа РАН
Россия, Республика Башкортостан, 450075 г. Уфа, пр. Октября, 141.

Тел.: +7 (347) 242 03 70.
**Email: olga.kiryanova27@gmail.com*

В работе представлено сравнение алгоритмов, применяемых при анализе нуклеотидных последовательностей (линейный поиск, алгоритм Кнута-Морриса-Пратта и алгоритм Бойера-Мура). Представлены результаты реализации алгоритма Бойера-Мура для поиска праймеров в последовательностях ДНК. Кроме того, описана методика однозначной ДНК-паспортизации на основе выявленного полиморфизма ДНК при анализе праймеров для полимеразной цепной реакции. Данная методика позволяет проводить однозначную ДНК-паспортизацию геномов, присваивая каждому из них штрих-код. Разработанная программа поиска праймеров реализована на языке программирования Python.

Ключевые слова: *паспортизация, штрих-код, ДНК полиморфизм, поиск образца в строке, алгоритм, Python, BioPython.*

Введение

Биоинформатика является одним из ключевых инструментов проведения научных исследований в области молекулярной биологии и генетики. Одним из типов исследуемых данных являются нуклеотидные последовательности. В цифровом формате информация о нуклеотидных последовательностях представляется в виде строк, состоящих из символов A (аденин), G (гуанин), C (цитозин) и T (тимин) для ДНК, вместо которого в РНК присутствует U (урацил). В общем случае эти задачи связаны с сравнением последовательностей, поиском подпоследовательностей, поиском сходства нескольких строк и т.д.

Одной из наиболее распространенных задач является поиск полного совпадения некоторой короткой строки, образца *A* в более длинной строке *T*. Необходимо найти все включения образца *A* в строке *T*. Эта задача схожа с поиском определенного слова в фрагменте текста.

Существует ряд алгоритмов, которые могут быть применены к решению данной задачи. Рассмотрим некоторые из них.

Алгоритмы поиска

Самым простым методом, который может быть использован в данном случае, является прямой поиск. Сравнение происходит слева направо. Левый конец образца находится на одной позиции с левым концом текста. «Сдвиг» образца на одну позицию происходит в случае несовпадения между образцом и строкой, или при полном совпадении всех элементов образца и строки (тогда необходимо обозначить вхождение образца в строке). Поиск продолжается до достижения конца строки. Сложность алгоритма можно оценить как $O(n \cdot m)$, где n – длина образца A, m – длина строки T.

Пример такого поиска представлен на *рис.* 1.

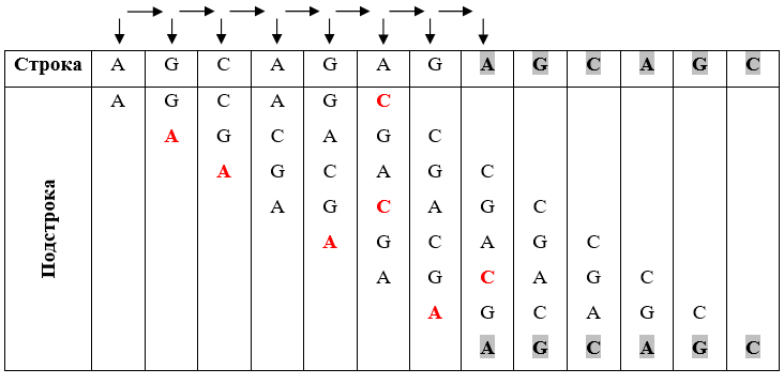


Рис. 1. Схематичное изображение прямого поиска AGCAGC в строке.
Красным цветом обозначены не совпавшие символы.

В данном случае проведено 22 сравнения символов образца и строки, прежде чем найдено полное совпадение. Очевидно, что данный метод неэффективен, особенно если речь идет о поиске в строках большого размера. В случае несовпадения происходит «откат» назад по строке, никак не учитываются проведенные сравнения. Кроме того, сдвиг по строке осуществляется только с единичным шагом.

Решить вышеописанную проблему можно увеличив «сдвиг» образца вдоль строки в случае несовпадения с элементами строки.

Одним из классических алгоритмов поиска образца в строке является алгоритм Кнута-Морриса-Пратта (КМП). Алгоритм КМП является самым известным алгоритмом с линейным временем для задачи точного совпадения образца со строкой [1]. Идея данного алгоритма заключается в том, чтобы максимально увеличить расстояния сдвига образца по строке, таким образом, сократив количество сравнений.

Алгоритм работает в два этапа. На первом этапе создается одномерный массив π, в котором хранятся значения для префикс-функции. Эти значения позволяют определить, на сколько позиций можно сдвинуть образец *A* вдоль строки *T*. На данном этапе происходит работа только с образцом. Чтобы создать массив π, необходимо найти префиксы образца равные ее суффиксам, то есть длина префикса должна быть равна длине суффикса [2].

Заполнение массива π происходит согласно правилу: префикс-функция для *i*-го символа возвращает значение, равное максимальной длине совпадающих префикса и суффикса подстроки в образце, которая заканчивается *i*-м символом. Для первого символа значение будет равное 0.

На втором этапе проводятся сравнения образца со строкой и, если символ в строке и соответствующий символ образца не совпали, то происходит соответствующий сдвиг образца *A* вдоль строки *T*. Сравниваются два символа, один из которых находится в строке, другой непосредственно в образце. Если символы совпадают, происходит проверка соседних справа символов, как показано на *рис.* 2, если же нет, то образ смещается на значение, соответствующее предшествующему не совпавшему элементу из массива π. Или же, можно произвести сдвиг образца таким образом, чтобы положение текущей позиции сравнения *j* совпадало с элементом образца, индекс которого равен значению $\pi[i]$.

В данном случае было проведено 16 сравнений символов образца и строки, прежде чем было найдено полное совпадение.

Эффективность алгоритма проявляется при частичном совпадении символов образца и строки. В случае алгоритма КМП сдвиг образца вдоль строки происходит не на один символ, как при прямом поиске, а на несколько [3]. Сложность алгоритма линейно зависит от объема входных данных и определяется как O(n+m), где n – длина образца, m – длина строки T [4].

Еще одним классическим алгоритмом поиска образца в строке является алгоритм Бойера-Мура (БМ). Он является самым быстрым алгоритмом среди известных классических алгоритмов общего назначения для поиска подстроки в строке или тексте [5].

Суть алгоритма БМ аналогична алгоритму КМП. Однако существенным различием является то, что просмотр совпадения ведется справа налево, другими словами, проверка начинается с последнего символа образца. Так же есть правило сдвига плохого символа, которое позволяет сдвигаться сразу на несколько позиций, значительно сократив время поиска [1].

Для начала строится таблица смещений для каждого символа образца согласно следующим правилам:

1) В таблицу смещения записывается расстояние от символа образца до крайнего правого символа;

2) Одинаковым символам в образце соответствует расстояние между крайним правым и последним символами;

3) Значение смещения для последнего символа соответствует длине образца, если он не встречался в образце левее.

i	0	1	2	3	4	5
Образец	A	G	C	A	G	C
Массив π[*i*]	0	0	0	1	2	3

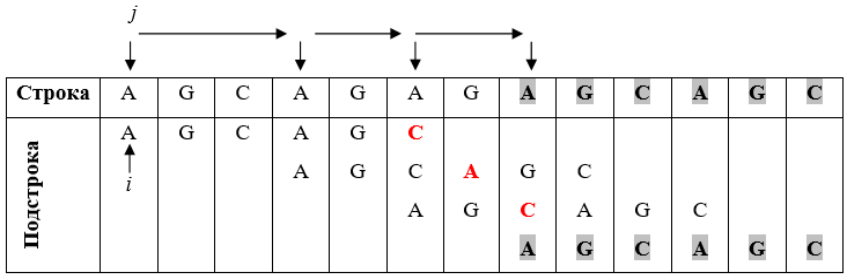


Рис.2. Схема поиска образца AGCAGC в строке согласно алгоритму Кнута-Морриса Пратта.
Красным цветом обозначены не совпавшие символы.

Затем начинается проверка последнего символа образца и соответствующего ему символа строки. Если последний символ образца *A* не совпадает с символом строки *T*, то образец *A* смещается на значение, соответствующее символу из строки *T* в *табл.* смещений. Если символы совпадают, то проверяется соседний (предпоследний) символ образца и так далее [6]. Совпадение всех символов образца *A* с наложенными символами строки *T* означает, что образец в строке найден. Весь алгоритм выполняется до тех пор, пока либо не будет найдено вхождение искомого образца, либо не будет достигнут конец строки.

Пример поиска по алгоритму БМ представлен на *рис.* 3.

В данном случае было проведено 13 сравнений символов образца и строки, прежде чем было найдено полное совпадение.

Проведя анализ принципов работы классических и наиболее часто применяемых алгоритмов общего назначения на примере алгоритмов КМП и БМ становится очевидным, что для достижения ускорения необходимо использовать множественные сдвиги, как на основе префикса и суффикса в алгоритме КМП, так и на основе правила сдвига плохого символа в алгоритме БМ. То есть необходима предварительная обработка искомого образца.

Проведя сравнительный анализ поиска образца в строках длины порядка 1 млн символов и небольшого по размеру алфавита (в нашем случае размер алфавита равен 5 (A, G, C, T, N – обозначается любой нуклеотид в каком-либо месте, когда он в силу разных причин неизвестен), было принято решение применить алгоритм БМ для поиска праймеров в нуклеотидной последовательности.

Постановка задачи

Полимеразная цепная реакция (ПЦР) – экспериментальный метод, который позволяет значительно увеличить концентрацию небольшого фрагмента ДНК в биологическом материале [7]. При проведении ПЦР важными компонентами реакционной смеси являются праймеры – короткие фрагменты, состоящие из 10–30 нуклеотидов [8]. На *рис.* 4 представлена схема проведения ПЦР.

i	0	1	2	3	4	5	(любой другой символ)
Образец	A	G	C	A	G	C	*
Сдвиг	2	1	3	2	1	3	6

j

Строка	A	G	C	A	G	A	G	**A**	**G**	**C**	**A**	**G**	**C**
Подстрока	A	G	C	A	G	**C**							
			A	G	C	A	G	**C**					
					A	G	**C**	A	G	C			
						A	G	C	A	G	**C**		
								A	**G**	**C**	**A**	**G**	**C**

i

Рис. 3. Схема поиска образца AGCAGC в строке согласно алгоритму Бойера-Мура. Красным цветом обозначены не совпавшие символы.

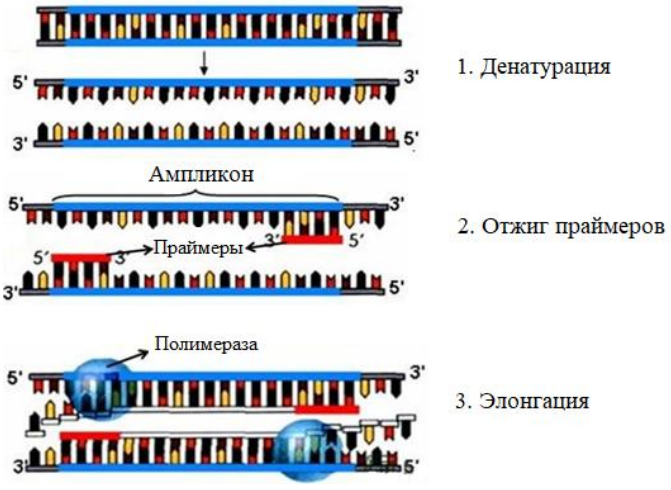


Рис. 4. Этапы проведения полимеразной цепной реакции.

Ранее нами был проведен анализ компьютерных программ, которые применяются для дизайна праймеров при проведении ПЦР [9]. Было рассмотрено более 100 программ, в которых можно проводить дизайн праймеров. Вручную провести анализ и подбор праймеров достаточно сложно и долго. Кроме того, удобно проводить предварительный компьютерный анализ прогностического характера, позволяющий определить положение праймеров, а также длины ампликонов (продуктов реакции). Такой подход позволяет получить реперные данные, из которых можно исходить, планируя тот или иной эксперимент и затем сравнивать с результатами натурного эксперимента. Разработанная методика позволяет определить положение конкретных праймеров, полученные длины ампликонов перевести в цифровой формат (или штрих-код).

При проведении экспериментов в области исследования сельскохозяйственных растений, проведении селекционных мероприятий актуальной проблемой является перевод полученных данных в цифровой формат. Цифровизация данных позволит каждому исследуемому геному (образцу, линии, сорту) присваивать определенного вида «паспорт». информацию о геномах растений достаточно легко каталогизировать, сортировать, хранить.

При проведении стандартной ПЦР используется одна пара праймеров для амплификации специфической последовательности. В мультиплексном методе ПЦР используется множество пар праймеров для амплификации многих последовательностей одновременно (обычно до 12 праймеров) [10]. И для этого необходимо найти все возможные комбинации, которые могут образовать праймеры между собой, определить места отжига на ДНК, размеры ампликонов. Если представить нуклеотидную последовательность в виде строки, а праймер в виде образца (короткой строки), то задачу можно сформулировать следующим образом. Необходимо найти позиции включения пары образцов в строке. Причем расстояние между образцами может варьировать в выбранном нами диапазоне от 51 до 500 символов. Данный диапазон определяется техническими возможностями оборудования для высокоточной детекции результатов ПЦР и целей натурного эксперимента [11].

Далее полученные длины ампликонов представлялись в виде полос, которые образовали штрих-код. Кроме того, полученные данные можно перевести в двоичный формат, в виде строки из 0 и 1 длиной 450 символов (соответствие диапазону от 51 до 500 нуклеотидов), где 1 – длины ампликонов, которые были найдены в геноме, 0 – отсутствующие ампликоны.

Стоит обратить внимание, что в качестве информации, которая однозначно определяет геном, являются данные о наборе (комплекте) мультиплексных праймеров, которые использовались при ПЦР, а также полученные размеры ампликонов. При другом наборе праймеров будет получен другой штрих-код.

Предложенная методика идентификации геномов действительно позволяет получить уникальный штрих-код. Возможное количество штрих-кодов для определенного количества ампликонов можно оценить как число сочетаний без повторения по формуле:

$$C_m^n = \frac{m!}{n!\,(m-n)!},$$

где m – диапазон исследуемых ампликонов (450), n – количество найденных ампликонов.

Согласно теории вероятностей самое большое количество штрих-кодов возникает при наличии половины ампликонов из анализируемого диапазона (225 из 450). Это количество составит 1.09e+134 штрих-кодов. Полученное число штрих-кодов с избытком достаточно для того, чтобы паспортизировать живые организмы. Даже в случае образования всего 20 ампликонов различных размеров число комбинаций составит порядка 10^{34}, что избыточно для ДНК-паспортизации сортов, сортообразцов, линий, пород, штаммов, рас. Предложенная методика паспортизации является универсальной для всех геномов, и может быть применена для любых других живых организмов, за исключением человека, поскольку для ДНК-идентификации личности в криминалистических целях используются другие подходы, наиболее перспективным из которых в плане цифровизации данных следует считать однонуклеотидный полиморфизм ДНК [12].

Программная реализация

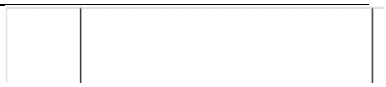
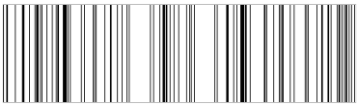
Программа поиска праймеров была реализована на языке программирования Python. Для удобства работы с нуклеотидными последовательностями применялись команды специализированной библиотеки BioPython.

Полученные данные позволили определить, какое количество ампликонов из выбранного диапазона длин возможно получить на геномах разного размера. Поиск проводился на 6 различных праймерах размером 10 нуклеотидов: AACCAGACAA, AAGGGACAAA, AACCGAACAA, AACGCACAAA, AAAACGCCAA, AACGCCAAAA.

Найденные размеры ампликонов были использованы для формирования штрих-кодов. Результаты представлены в *табл.* 1.

Таблица 1

Выходные данные для тестовых геномов

Геном	Праймеры	Размеры ампликонов	Штрих-код
Arabidopsis thaliana (L.) Heynh 130 000 000 п.н.	AACCAGACAA AAGGGACAAA AACCGAACAA	112, 355, 383, 467	
Solanum tuberosum L. 1 000 000 п.н.	AACGCACAAA AAAACGCCAA AACGCCAAAA	74, 94, 168, 180, 182, 183, 184, 185, 188, 195, 209, 254, 258, 265, 266, 268, 269, 274, 275, 298, 299, 302, 304, 305, 319, 345, 347, 348, 353, 361, 365, 399, 402, 404, 415, 427, 449, 465, 467, 472, 484	
Triticum aestivum L. 17 000 000 000 п.н.		51, 55, 56, 66, 75, 76, 77, 84, 91, 93, 94, 95, 98, 100, 106, 113, 118, 120, 121, 127, 128, 129, 130, 131, 133, 136, 150, 154, 165, 167, 169, 180, 187, 188, 196, 202, 208, 211, 238, 242, 250, 254, 255, 256, 258, 259, 263, 268, 274, 284, 288, 290, 294, 320, 323, 335, 336, 337, 344, 348, 353, 354, 355, 356, 357, 359, 360, 365, 383, 384, 386, 395, 402, 404, 406, 407, 416, 421, 435, 437, 439, 450, 456, 457, 464, 465, 468, 476, 478, 479, 456, 482, 485, 487, 488, 491, 494, 498	

Следует отметить, что для геномов размера более 1 миллиарда п.н. количество выявленных ампликонов достаточно для того, чтобы проводить сравнение на уровне родственных сортообразцов. Для геномов меньших размеров такое сравнение не позволит получить объективные результаты, т.к. в этом случае выявляется недостаточное количество полос в штрих-коде и требуется использование большего числа праймеров в мультиплексной ПЦР.

Разработанная методика паспортизации позволяет получать именно цифровую информацию о геноме, проводить легкое и наглядное сравнение штрих-кодов, решая проблему сравнения экспериментальных данных, исключая человеческий фактор [13].

Заключение

В работе проведен анализ классических алгоритмов поиска образца в строке. Реализована программа поиска праймеров в нуклеотидных последовательностях. Разработанная программа, позволяет, не проводя натурного эксперимента, протестировать несколько праймеров, а также получить представление об успешности проведения натурного эксперимента. Найденные размеры ампликонов позволяют формировать штрих-код, который в совокупности с применяемым набором праймеров однозначно идентифицирует конкретный геном. Полученные данные служат в качестве реперных данных при планировании лабораторного эксперимента. В дальнейшем планируется проведение натурных экспериментов для сравнения с результатами компьютерного анализа.

Работа выполнена при финансовой поддержке гранта РФФИ 17-44-020120.

ЛИТЕРАТУРА

1. Гасфилд Д. Строки, деревья и последовательности в алгоритмах: Информатика и вычислительная биология. СПб.: Невский Диалект, БХВ-Петербург, 2003. С. 654.
2. Префикс-функция. Алгоритм Кнута-Морриса-Пратта. URL: https://brestprog.by/topics/prefixfunction/
3. Смит Б. Методы и алгоритмы вычислений на строках. М.: Вильямс, 2006. С. 496.
4. Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы: построение и анализ / под ред. И. В. Красикова. М.: Вильямс, 2005. С. 1296.
5. Боровский А. Алгоритмы поиска в тексте. RSDN Magazine. URL: https://rsdn.org/article/alg/textsearch.xml.
6. Упрощенный алгоритм Бойера-Мура. URL: https://habr.com/ru/post/116725/
7. Глик Б., Пастернак Дж. Молекулярная биотехнология. Принципы и применение. М.: Мир, 2002. С. 589.
8. Гарафутдинов Р. Р., Баймиев А. Х., Малеев Г. В., Алексеев Я. И., Зубов В. В., Чемерис Д. А., Кирьянова О. Ю., Губайдуллин И. М., Матниязов Р. Т., Сахабутдинова А. Р., Никоноров Ю. М., Кулуев Б. Р., Баймиев А. Х., Чемерис А. В. Разнообразие праймеров для ПЦР и принципы их подбора // Биомика. 2019. Т 11. №1. С. 23–70.
9. Чемерис Д. А., Кирьянова О. Ю., Губайдуллин И. М., Чемерис А. В. Дизайн праймеров для полимеразной цепной реакции (краткий обзор комп. прогр. и баз данных) // Биомика. 2016. Т. 8. №3. С. 215–238.
10. Markoulatos P., Siafakas N., Moncany M. Multiplex polymerase chain reaction: A practical approach // J. Clin. Lab. Anal. 2002. No 16. Pp. 47–51.
11. Кирьянова О. Ю., Чемерис А. В. Моделирование поиска праймеров в цепи ДНК // Тр. международ. конф. ИТНТ-2019. Самара. 2019. С. 774–778.
12. Чемерис Д. А., Сагитов А. М., Аминев Ф. Г., Луценко В. И., Гарафутдинов Р. Р., Сахабутдинова А. Р., Василов Р. Г., Алексеев Я. И., Сломинский П. А., Хуснутдинова Э. К., Чемерис А. В. Эволюция подходов к ДНК-идентификации личности // Биомика. 2018. Т. 10. №1. С. 85–140.
13. Jiang B., Zhao Y., Yi H., Huo Y., Wu H., Ren J., Ge J., Zhao J., Wang F. PIDS: A User-Friendly Plant DNA Fingerprint Database Management System // Genes. 2020, Vol 11. No 4. P. 373.

Поступила в редакцию 19.05.2020 г.

DOI: 10.33184/bulletin-bsu-2020.2.10

SEARCH ALGORITHMS IN THE ANALYSIS OF NUCLEOTIDE SEQUENCES FOR UNAMBIGUOUS IDENTIFICATION OF GENOMES

© O. Yu. Kiryanova[1]*, L. U. Akhmetzianova[1,2], I. M. Gubaydullin[1,2]

[1]Ufa State Petroleum Technological University
1 Kosmonavtov Street, 450064 Ufa, Republic of Bashkortostan, Russia.

[2]Institute of Petrochemistry and Catalisys, Ufa Scientific Center of RAS
141 Oktyabrya Avenue, 450075 Ufa, Republic of Bashkortostan, Russia.

Phone: +7 (347) 242 03 70.
**Email: olga.kiryanova27@gmail.com*

In the paper, the comparison of algorithms that are used in the analysis of nucleotide sequences (linear search, the Knuth-Morris-Pratt algorithm, and the Boyer-Moore algorithm) was discussed. The results of the Boyer-Moore algorithm application to search for primers in DNA sequences were presented. An unambiguous DNA certification method based on detected DNA polymorphism in primers analysis for the polymerase chain reaction was proposed. A new software for search of primers was developed using Python language with BioPython library. The proposed approach is based on unique barcode that identifies a particular organism. The studies were conducted using several types of crops and the model plant (*Solanum tuberosum, Triticum aestivum, Arabidopsis thaliana*). Many approaches are used for DNA certification of plant varieties but none of them provides unambiguous digital data. Thus, the suggested approach for DNA certification (cataloging)/identification of living organisms is unique; without conducting a full-scale experiment, it is possible to test several primers as well as get an idea of the full-scale experiment success. The uniqueness of the proposed technique is that allows a researcher to systematize data for different primers and DNA sequences without taking into account their natural affiliation. Thus, the generated information is a kind of digital passport for varieties, breeds, races, strains of various organisms.

Keywords: pattern search in a string, algorithm, Python, BioPython, certification, barcode, Numba.

Published in Russian. Do not hesitate to contact us at bulletin_bsu@mail.ru if you need translation of the article.

REFERENCES

1. Gusfield D. Stroki, derev'ya i posledovatel'nosti v algoritmakh: Informatika i vychislitel'naya biologiya [Strings, trees, and sequences in algorithms: Computer science and computational biology]. Saint Petersburg: Nevskii Dialekt, BXV-Peterburg, 2003. Pp. 654.
2. Prefiks-funktsiya. Algoritm Knuta-Morrisa-Pratta. URL: https://brestprog.by/topics/prefixfunction/
3. Smith B. Metody i algoritmy vychislenii na strokakh [Methods and algorithms for computing on strings]. Moscow: Vil'yams, 2006. Pp. 496.
4. Cormen T., Leiserson Ch., Rivest R., Stein C. Algoritmy: postroenie i analiz [Introduction to algorithms]. Ed. I. V. Krasikova. Moscow: Vil'yams, 2005. Pp. 1296.
5. Borovskii A. Algoritmy poiska v tekste. RSDN Magazine. URL: https://rsdn.org/article/alg/textsearch.xml.
6. Uproshchennyi algoritm Boiera-Mura. URL: https://habr.com/ru/post/116725/
7. Glick B., Pasternak J. Molekulyarnaya biotekhnologiya. Printsipy i primenenie [Molecular biotechnology. Principles and application]. Moscow: Mir, 2002. Pp. 589.
8. Garafutdinov R. R., Baimiev A. Kh., Maleev G. V., Alekseev Ya. I., Zubov V. V., Chemeris D. A., Kir'yanova O. Yu., Gubaidullin I. M., Matniyazov R. T. Biomika. 2019. T 11. No. 1. Pp. 23–70.
9. Chemeris D. A., Kir'yanova O. Yu., Gubaidullin I. M., Chemeris A. V. Biomika. 2016. Vol. 8. No. 3. Pp. 215–238.
10. Markoulatos P., Siafakas N., Moncany M. J. Clin. Lab. Anal. 2002. No 16. Pp. 47–51.
11. Kir'yanova O. Yu., Chemeris A. V. Tr. mezhdunarod. konf. ITNT-2019. Samara. 2019. Pp. 774–778.
12. Chemeris D. A., Sagitov A. M., Aminev F. G., Lutsenko V. I., Garafutdinov R. R. Biomika. 2018. Vol. 10. No. 1. Pp. 85–140.
13. Jiang B., Zhao Y., Yi H., Huo Y., Wu H., Ren J., Ge J., Zhao J., Wang F. Genes. 2020, Vol 11. No 4. Pp. 373.

Received 19.05.2020.